MPLS Enhancements to the Interfaces MIB

This document describes the Multiprotocol Label Switching (MPLS) enhancements to the existing interfaces Management Information Base (MIB) (RFC 2233) to support an MPLS layer. This layer provides counters and statistics specifically for MPLS.

History for the MPLS Enhancements to the Interfaces MIB Feature

Release	Modification
12.0(23)S	This feature was introduced.
12.2(18)S	This feature was integrated into Cisco IOS Release 12.2(18)S.
12.2(25)S	This feature was updated to work with MPLS High Availability on the Cisco 7500 series routers.
12.2(27)SBC	This feature was integrated into Cisco IOS Release 12.2(27)SBC.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information about MPLS Enhancements to Interfaces MIB

The Interfaces MIB (IF MIB) provides a Simple Network Management Protocol (SNMP)-based method for managing interfaces. Each entry in the IF MIB establishes indexing, statistics, and stacking relationships among underlying physical interfaces, subinterfaces, and Layer 2 protocols that exist within Cisco IOS.

The enhancements add an MPLS layer to the IF MIB as a Layer 2 protocol to provide statistics for traffic encapsulated as MPLS on an interface. In this new structure, MPLS-specific data, such as MPLS-encapsulated traffic counters and the MPLS maximum transmission unit (MTU), resides on top of the underlying physical or virtual interface to allow separation from non-MPLS data.

The enhancements also allow you to view indexing, statistics, and stacking relationships using the ifStackTable. MPLS layer interfaces are stacked above the underlying physical or virtual interface that is actually forwarding the MPLS traffic. MPLS traffic engineering tunnels are then stacked above those MPLS layers.

The IF MIB supports several types of interfaces. A virtual interface that provides protocol statistics for MPLS-encapsulated traffic has been added. This new interface is stacked above real Cisco IOS interfaces or subinterfaces, such as Ethernet (et0) or ATM (at1/1.1).

When any MPLS application starts, Cisco IOS creates a corresponding MPLS layer above each interface capable of supporting MPLS.

An IF MIB entry is created when you enable any MPLS application on an interface; the entry is removed when you disable the MPLS application.

MIB Tables

The IF MIB consists of the following tables:

- ifTable—Contains information on each interface in the network. Its definition of an interface includes any sublayers of the internetwork layer of the interface. MPLS interfaces fit into this definition of an interface. Therefore, each MPLS-enabled interface is represented by an entry in the ifTable. (See Table 1.)
- ifXTable—Contains objects that have been added to the IF MIB as a result of the interface evolution effort or replacements for objects from the original MIB-II ifTable. This table also contains objects that were previously in the ifExtsTable. (See Table 2.)
- ifStackTable—Contains objects that define the relationships among the sublayers of an interface. (See Table 3.)
- ifRcvAddressTable—Contains objects that define the media-level addresses received by an interface. (See Table 4.)

Note There are objects that are not supported by some interface types. The corresponding MPLS layer objects might not be supported if they depend on an unsupported object at the underlying layer.

The notation used in the IF MIB follows the conventions defined in Abstract System Notation One (ASN.1). ASN.1 defines an Open System Interconnection (OSI) language used to describe data types independently from particular computer structures and presentation techniques. Each object in the MIB incorporates a DESCRIPTION field that includes an explanation of the object's meaning and usage, which, together with the other characteristics of the object (SYNTAX, MAX-ACCESS, and INDEX) provides sufficient information for management application development, as well as for documentation and testing.

A network administrator can access the entries (objects) in the IF MIB by means of any SNMP-based network management system (NMS). The network administrator can retrieve information in the IF MIB using standard SNMP **get** and **getnex**t operations.

ifTable Objects

Table 1 lists the ifTable objects and their descriptions.

Table 1 ifTable Objects and Descriptions

Object	**Description**
ifIndex	A unique value, greater than 0, for each interface.
ifDescr	A textual string containing information about the MPLS layer. <*underlying interface name*>-mpls layer
ifType	The kind of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA) through updating the syntax of the IANAifType textual convention. MPLS-layer entries are type mpls (166).
ifMtu	The size of the largest MPLS-encapsulated packet that can be sent or received on the interface, specified in octets. By default, this object is the same as the underlying physical or virtual interface, but can be configured by use of the **mpls mtu** command.
ifSpeed	An estimate of the interface's current bandwidth in bits per second. This object is the same as the underlying physical or virtual interface; however, it has no MPLS-specific value.
ifPhysAddress	The interface's address at its protocol sublayer. This object is the same as the underlying physical or virtual interface.
ifAdminStatus	The administrative state of the underlying physical or virtual interface. This object is the same as the underlying physical or virtual interface.
ifOperStatus	The operational state of MPLS on the interface. Use the **show mpls interface** command to display this information. **Note** MPLS can be in the up or down state when the underlying physical or virtual interface is up. However, when the underlying physical or virtual interface is down, MPLS is always in the down state.

Table 1 ifTable Objects and Descriptions (continued)

Object	Description
ifLastChange	The value of sysUpTime when the interface entered its current MPLS-specific operational state.
ifInOctets	The total number of MPLS-encapsulated octets received on the interface.
ifInUcastPkts	The number of MPLS-encapsulated packets received on the underlying physical or virtual interface.
ifInNUcastPkts	The number of packets, delivered by this sublayer to a higher sublayer, which were addressed to a multicast or broadcast address at this sublayer. This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.
ifInDiscards	The number of inbound packets selected to be discarded, even though no errors had been detected, to prevent delivery to a higher-layer protocol. One reason for discarding such a packet is to free up buffer space. This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.
ifInErrors	The number of inbound packets that contained errors preventing them from being delivered to a higher-layer protocol. This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.
ifInUnknownProtos	The number of packets that were discarded because of an unknown or unsupported protocol received via the interface. This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.
ifOutOctets	The total number of MPLS-encapsulated octets transmitted from the interface.
ifOutUcastPkts	The number of MPLS-encapsulated packets transmitted from the underlying physical or virtual interface.
ifOutNUcastPkts	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent. This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.

Table 1 ifTable Objects and Descriptions (continued)

Object	Description
ifOutDiscards	The number of outbound packets selected to be discarded even though no errors had been detected to prevent their transmission. One reason for discarding such a packet is to free up buffer space. This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.
ifOutErrors	The number of outbound packets that could not be transmitted because of errors. This object is not supported on the MPLS layer because the event occurs on the underlying physical or virtual interface where it is counted.
ifOutQLen	The length of the output packet queue (in packets). This object is the same as the underlying physical or virtual interface. It is not supported for MPLS.
ifSpecific	A reference to MIB definitions specific to the particular media being used to recognize the interface. This object is not supported because there is no such reference for MPLS.

ifXTable Objects

Table 2 lists the ifXTable objects and their descriptions.

Note High-capacity (HC) objects supported in the IF MIB require SNMPv2c or higher.

Table 2 ifXTable Objects and Descriptions

Object	Definition
ifName	The textual name of the underlying physical or virtual interface.
ifInMulticastPkts	The number of packets that were addressed to a multicast address at this sublayer and that were then delivered by this sublayer to a higher sublayer. This object is not supported because MPLS currently does not support multicast.
ifInBroadcastPkts	The number of packets that were addressed to a broadcast address at this sublayer and delivered by this sublayer to a higher sublayer. This object is not supported because MPLS currently does not support broadcast.

Table 2 ifXTable Objects and Descriptions (continued)

Object	Definition
ifOutMulticastPkts	The total number of packets that higher-level protocols requested be transmitted and that were addressed to a multicast address at this sublayer, including those that were discarded or not sent. This object is not supported because MPLS currently does not support multicast.
ifOutBroadcastPkts	The total number of packets that higher-level protocols requested be transmitted and that were addressed to a broadcast address at this sublayer, including those that were discarded or not sent. This object is not supported because MPLS currently does not support broadcast.
ifHCInOctets	The total number of MPLS-encapsulated octets received on the interface. This object is a 64-bit version of ifInOctets.
ifHCInUcastPkts	The number of MPLS-encapsulated packets received on the underlying physical or virtual interface. This object is a 64-bit version of ifInUcastPkts.
ifHCInMulticastPkts	The number of packets that were addressed to a multicast address at this sublayer and delivered by this sublayer to a higher sublayer. This object is not supported because MPLS currently does not support multicast. This object is a 64-bit version of ifInMulticastPkts.
ifHCInBroadcastPkts	The number of packets that were addressed to a broadcast address at this sublayer and delivered by this sublayer to a higher sublayer. This object is not supported because MPLS currently does not support broadcast. This object is a 64-bit version of ifInBroadcastPkts.
ifHCOutOctets	The total number of MPLS-encapsulated octets transmitted from the interface. This object is a 64-bit version of ifOutOctets.
ifHCOutUcastPkts	The number of MPLS-encapsulated packets transmitted from the underlying physical or virtual interface. This object is a 64-bit version of ifOutUcastPkts.
ifHCOutMulticastPkts	The total number of packets that higher-level protocols requested be transmitted and that were addressed to a multicast address at this sublayer, including those that were discarded or not sent. This object is not supported because MPLS currently does not support multicast. This object is a 64-bit version of ifOutMulticastPkts.

Table 2 ifXTable Objects and Descriptions (continued)

Object	Definition
ifHCOutBroadcastPkts	The total number of packets that higher-level protocols requested be transmitted and that were addressed to a broadcast address at this sublayer, including those that were discarded or not sent. This object is not supported because MPLS currently does not support broadcast. This object is a 64-bit version of ifOutBroadcastPkts.
ifLinkUpDownTrapEnable	Indicates whether LinkUp/LinkDown traps should be generated for this interface. Link up/down traps are not supported in this release; consequently, this object is set to disabled(2) for MPLS.
ifHighSpeed	An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. This object is the same as the underlying physical or virtual interface; however, it has no MPLS-specific value.
ifPromiscuousMode	Has a value of false(2) if this interface accepts only packets and frames that are addressed to this station; has a value of true(1) if the station accepts all packets and frames transmitted on the media. This object is the same as the underlying physical or virtual interface.
ifConnectorPresent	Has the value true(1) if the interface sublayer has a physical connector and the value false(2) otherwise. For MPLS, the value is always false(2).
ifAlias	Another name for an interface as specified by a network manager. This object is the same as the underlying physical or virtual interface.
ifCounterDiscontinuityTime	The value of sysUpTime on the most recent occasion at which any one or more of this interface’s MPLS-specific counters experienced an interruption in incrementing.

ifStackTable Objects

Table 3 lists the ifStackTable objects and their descriptions.

Table 3 ifStackTable Objects and Definitions

Object	Definition
ifStackHigherLayer	The value of ifIndex corresponding to the higher sublayer of the relationship; that is, the sublayer that runs on top of the sublayer identified by the corresponding instance of the ifStackLowerLayer. **Note** Index objects are not accessible in a MIB walk. This value is part of the object identifier (OID) for every object in the ifStackTable.
ifStackLowerLayer	The value of ifIndex corresponding to the lower sublayer of the relationship; that is, the sublayer that runs below the sublayer identified by the corresponding instance of the ifStackHigherLayer. **Note** Index objects are not accessible in a MIB walk. This value is part of the OID for every object in the ifStackTable.
ifStackStatus	Used to create and delete rows in the ifStackTable; the status is always active(1) for MPLS.

ifRcvAddressTable Objects

Table 4 lists the ifRcvAddressTable objects and their descriptions.

Note Entries for the MPLS layer do not appear in the ifRcvAddressTable.

Table 4 ifRcvAddressTable Objects and Descriptions

Object	Definition
ifRcvAddressAddress	An address for which the system accepts packets and frames on this entry's interface. **Note** Index objects are not accessible in a MIB walk. This value is part of the OID for every object in the ifRcvAddressTable.
ifRcvAddressStatus	Used to create and delete rows in the ifRcvAddressTable.
ifRcvAddressType	Type of storage used for each entry in the ifRcvAddressTable.

Scalar Objects

The IF MIB supports the following scalar objects:

- ifStackLastChange—The value of sysUpTime at the time of the last change of the entire interface stack. A change of the interface stack is defined to be any creation, deletion, or change in value of any instance of ifStackStatus. If the interface stack has been unchanged since the last reinitialization of the local network management subsystem, then this object contains a 0 value.
- ifTableLastChange—The value of sysUpTime at the time of the last creation or deletion of an entry in the ifTable. If the number of entries has been unchanged since the last reinitialization of the local network management subsystem, then this object contains a 0 value.

Stacking Relationships for MPLS Layer Interfaces

The ifStackTable within the IF MIB provides a conceptual stacking relationship between the interfaces and subinterfaces represented as entries in the ifTable.

The ifStackTable is indexed like a linked list. Each entry shows a relationship between two interfaces providing the ifIndexes of the upper and the lower interface. The entries chain together to show the entire stacking relationship. Each entry links with one another until the stack terminates with an ifIndex of 0 at the highest and lowest ends of the stack. For example, in Figure 1, the indexes .10.5 show that ifIndex 10 is stacked upon ifIndex 5. There are 0 entries at the highest and lowest ends of the stack; in Figure 1, the indexes .0.15 and .72.0 are the highest and lowest ends of the stack, respectively.

Figure 1 Sample ATM Stacking Relationship in the ifStackTable

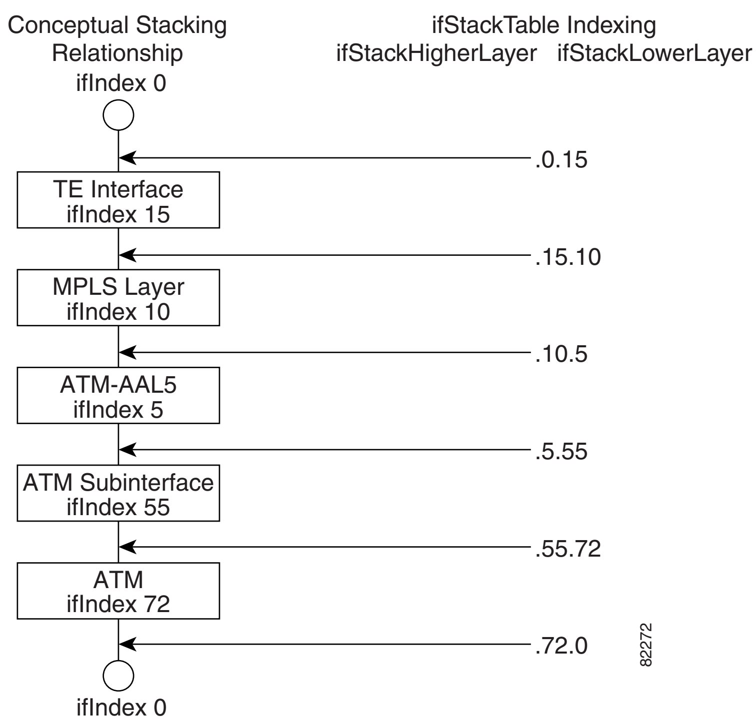


Table 5 describes the indexing of the ifStackTable for the layer relationships shown in Figure 1.

Note The order of the entries in Table 5 might not be the same as that seen in the MIB walk, which has to follow SNMP ordering rules.

Table 5 Layer Relationships

Layer Relationship (in Descending Order)	ifStackHigherLayer/ifStackLowerLayer
Traffic engineering (TE) interface as top layer	.0.15
TE interface stacked upon MPLS layer	.15.10
MPLS layer stacked upon ATM-AAL5	.10.5
ATM-AAL5 layer stacked upon ATM subinterface	.5.55
ATM subinterface stacked upon ATM	.55.72
ATM as bottom layer	.72.0

Stacking Relationships for Traffic Engineering Tunnels

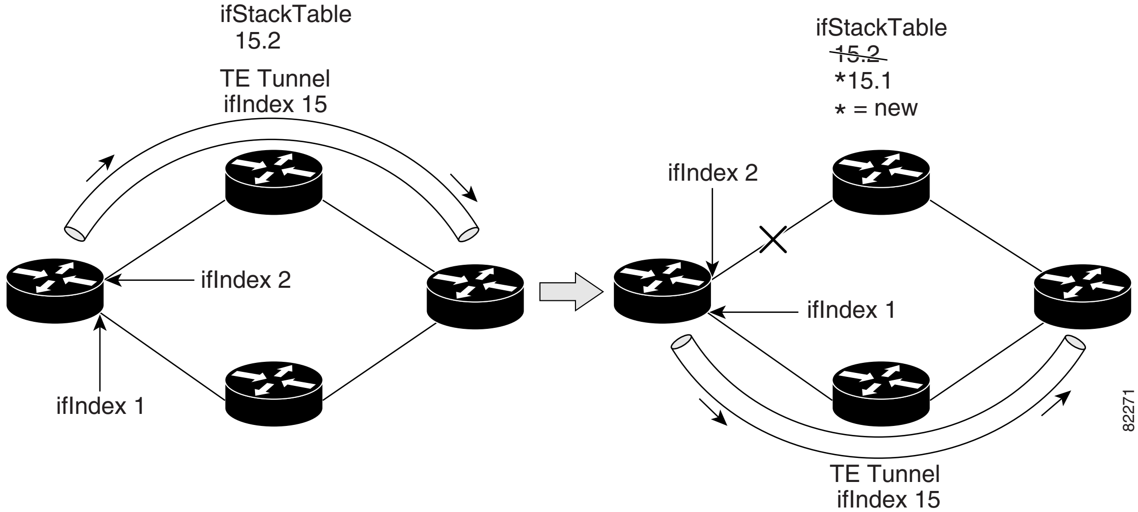
MPLS TE tunnels are represented in Cisco IOS and the IF MIB as virtual interfaces. When properly signaled, TE tunnels pass traffic through MPLS over a physical interface. This process dictates that a TE tunnel is to be stacked on an MPLS layer that is stacked on an underlying interface.

TE tunnels can also change paths in response to different error or network conditions. These changes are instigated by use of the RSVP-TE signaling protocol. When a change occurs, a tunnel can switch to a different MPLS interface. If no signaling path exists, no paths will be chosen and thus no MPLS interface will be used.

Because a TE tunnel is represented as an IF MIB ifTable entry, the ifStackTable also contains an entry corresponding to the TE tunnel. If the TE tunnel is successfully signaled, the ifStackTable also contains a link between the tunnel interface and one MPLS interface. Because it is possible for a TE tunnel to not have a corresponding signaled path, it is possible for a TE tunnel's ifStackTable entry to not have a corresponding lower layer. In this case, the lower layer variable contains the value of 0.

Figure 2 shows a TE tunnel before (left) and after (right) being rerouted and the effect on the ifStackTable. When ifIndex 2 fails, the TE tunnel is rerouted through ifIndex1, the 15.2 entry is removed from the ifStackTable, and the 15.1 entry is added.

Figure 2 Sample TE Tunnel Stacking Relationship



MPLS Label Switching Router MIB Enhancements

All of the ifIndex references in the MPLS-LSR-MIB tables have changed from the ifIndex of the underlying physical or virtual interface to the ifIndex of the MPLS layer.

Table 6 shows the specific changes.

Table 6 MPLS-LSR-MIB ifIndex Objects Enhanced

Table	ifIndex
MPLS interface configuration table (mplsInterfaceConfTable)	mplsInterfaceConfIndex
MPLS in-segment table (mplsInSegmentTable)	0[1]
MPLS cross-connect table (mplsXCTable)	0[2]
MPLS out-segment table (mplsOutSegmentTable)	mplsOutSegmentIfIndex

1. The LSR MIB uses a 0 platform-wide interface for all of the label tables. See the *MPLS Label Switching Router MIB* for more information.
2. The LSR MIB uses a 0 platform-wide interface for all of the label tables. See the *MPLS Label Switching Router MIB* for more information.

The following objects from the mplsInterfaceConfTable are affected:

- mplsInterfaceOutPackets—Count only MPLS-encapsulated packets
- mplsInterfaceInPackets—Count only MPLS-encapsulated packets

Benefits

Improved Accounting Capability

By viewing the MPLS layer, you get MPLS-encapsulated traffic counters that do not include non-MPLS encapsulated traffic (for example, IP packets). Therefore, the counters are more useful for MPLS-related statistics.

TE Tunnel Interfaces

For TE tunnel interfaces, the stacking relationship reflects the current underlying MPLS interface that is in use and dynamically changes as TE tunnels reoptimize and reroute.

MPLS-Specific Information

The MPLS layer shows MPLS-specific information including the following:

- If MPLS is enabled
- MPLS counters
- MPLS MTU
- MPLS operational status

Restrictions

- Link up and link down traps for the MPLS layer are not supported in this release.
- Write capability using the SNMP SET command is not supported for the MPLS layer in this release.
- Some counters, including discard and multicast, increment on the underlying physical layer; therefore, they equal 0 because they never reach the MPLS layer.

How to Configure MPLS Enhancements to Interfaces MIB

See the following sections for configuration tasks for the MPLS Enhancements to Interfaces MIB feature. Each task in the list is identified as either optional or required.

- Enabling the SNMP Agent (required)
- Verifying That the SNMP Agent Has Been Enabled (optional)

Prerequisites

The MPLS Enhancements to the Interfaces MIB feature requires the following:

- SNMP must be installed and enabled on the label switch routers (LSRs).
- MPLS must be enabled on the LSRs.
- An MPLS application must be enabled on an interface.

Enabling the SNMP Agent

The SNMP agent for the IF MIB is disabled by default. To enable the SNMP agent, use the commands in the following sections.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **configure terminal**
4. **snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*]
5. **end**
6. **write memory**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** **Example:** `Router# show running-config`	Displays the running configuration of the router to determine if an SNMP agent is already running on the device. If no SNMP information is displayed, continue with the next step. If any SNMP information is displayed, you can modify the information or change it as desired.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro**] [*number*] **Example:** `Router(config)# snmp-server community public ro`	Configures read-only (ro) SNMP community strings.
Step 5	**end** **Example:** `Router(config)# end`	Exits to privileged EXEC mode.
Step 6	**write memory** **Example:** `Router# write memory`	Writes the modified SNMP configuration into NVRAM of the router, permanently saving the SNMP settings.

Verifying That the SNMP Agent Has Been Enabled

To verify that the SNMP agent has been enabled, perform the following steps:

Step 1 Access the router through a Telnet session:

```
Prompt# telnet xxx.xxx.xxx.xxx
```

where *xxx.xxx.xxx.xxx* represents the IP address of the target device.

Step 2 Enter privileged EXEC mode:

```
Router# enable
```

Step 3 Display the running configuration and look for SNMP information:

```
Router# show running-configuration
...
...
snmp-server community public RO
```

If you see any snmp-server statements, SNMP has been enabled on the router.

Configuration Examples for MPLS Enhancements to Interfaces MIB

The following example shows how to enable an SNMP agent:

```
Router# configure terminal

Router(config)# snmp-server community
```

In the following example, SNMPv1 and SNMPv2C are enabled. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string public.

```
Router(config)# snmp-server community public
```

In the following example, read-only access is allowed for all objects to members of access list 4 that specify the comaccess community string. No other SNMP managers have access to any objects.

```
Router(config)# snmp-server community comaccess ro 4
```

Additional References

The following sections provide references related to MPLS Enhancements to Interfaces MIB.

Related Documents

Related Topic	Document Title
Configuring SNMP using Cisco IOS software	• *Cisco IOS Configuration Fundamentals Configuration Guide,* Release 12.2 • *Cisco IOS Configuration Fundamentals Command Reference,* Release 12.2 • *MPLS Label Switching Router MIB*

Standards

Standards	Title
draft-ietf-mpls-lsr-mib-08.txt	*Multiprotocol Label Switching (MPLS) Label Switch Router (LSR) Management Information Base*

MIBs

MIBs	MIBs Link
Interfaces Group MIB (IF MIB)	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2233	*The Interfaces Group MIB using SMIv2*
RFC 1229	*Extensions to the Generic-Interface MIB*
RFC 1213	*Management Information Base for Network Management of TCP/IP-based internets: MIB-II*
RFC 1157	*A Simple Network Management Protocol (SNMP)*
RFC 1156	*Management Information Base for Network Management of TCP/IP-based internets*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.cm/techsupport

Command Reference

This feature uses no new or modified commands.

Glossary

ATM—Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media, such as E3, SONET, and T3.

ATM-AAL5—ATM adaptation layer 5. One of four AALs recommended by the ITU-T. AAL5 supports connection-oriented variable bit rate (VBR) services and is used predominantly for the transfer of classical IP over ATM and LAN emulation (LANE) traffic. AAL5 uses simple and efficient AAL (SEAL) and is the least complex of the current AAL recommendations. It offers low bandwidth overhead and simpler processing requirements in exchange for reduced bandwidth capacity and error-recovery capability.

encapsulation—Wrapping of data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit. Also, when dissimilar networks are bridged, the entire frame from one network is simply placed in the header used by the data link layer protocol of the other network.

IETF—Internet Engineering Task Force. A task force (consisting of more than 80 working groups) that is developing standards for the Internet and the IP suite of protocols.

interface—The boundary between adjacent layers of the ISO model.

label—A short, fixed-length identifier that is used to determine the forwarding of a packet.

label switching—The forwarding of IP (or other network layer) packets by means of a label-swapping algorithm based on network layer routing algorithms. The forwarding of these packets uses the exact match algorithm and rewrites the label.

LSR—label switch router. A device that forwards MPLS packets based on the value of a fixed-length label encapsulated in each packet.

MIB—Management Information Base. A database of network management information that is used and maintained by a network management protocol such as SNMP. The value of a MIB object can be changed or retrieved by means of SNMP commands, usually through a network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

MPLS—Multiprotocol Label Switching. A switching method that forwards IP traffic through use of labels. Each label instructs the routers and the switches in the network where to forward packets based on preestablished IP routing information.

MPLS interface—An interface on which MPLS traffic is enabled.

MTU—Maximum transmission unit. Maximum packet size, in bytes, that a particular interface can handle.

NMS—network management system. A system responsible for managing at least part of a network. An NMS is generally a reasonably powerful and well-equipped computer, such as an engineering workstation. NMSs communicate with agents to help keep track of network statistics and resources.

OID—Object identifier. Values are defined in specific MIB modules. The Event MIB allows you or an NMS to watch over specified objects and to set event triggers based on existence, threshold, and Boolean tests. An event occurs when a trigger is fired; this means that a specified test on an object returns a value of true. To create a trigger, you or an NMS configures a trigger entry in the mteTriggerTable of the Event MIB. This trigger entry specifies the OID of the object to be watched. For each trigger entry type, corresponding tables (existence, threshold, and Boolean tables) are populated with the information required for carrying out the test. The MIB can be configured so that, when triggers are activated (fired), either an SNMP **set** is performed, a notification is sent out to the interested host, or both.

SNMP—Simple Network Management Protocol. A management protocol used almost exclusively in TCP/IP networks. SNMP provides a means for monitoring and controlling network devices, and for managing configurations, statistics collection, performance, and security.

traffic engineering tunnel—A label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing could cause the tunnel to take.

trap—A message sent by an SNMP agent to a network management station, console, or terminal, indicating that a significant event occurred. Traps are less reliable than notification requests, because the receiver does not send an acknowledgment when it receives a trap. The sender cannot determine if the trap was received.

tunnel—A secure communication path between two peers, such as routers.